

## 활 동 보 고 서

회차	9주차	지도교수 확인란	
일시	2019.10.07.(월) 18:00~22:00		
장소	충남대 경상대 스터디룸 1층		
주제	파이썬으로 거북이 대포게임 만들기		
학습 내용 요약			

<거북이 대포게임>

거북이 대포게임은 위, 아래 키보드 방향키로 발사각도를 조절하고 스페이스바로 대포를 발사해서 화살촉모양의 대포가 초록색 목표지점에 떨어뜨려야하는 게임이다. 초록색 목표지점에 떨어지면 Good!이라는 메시지가, Bad!라는 메시지가 보여 지지만 우리는 조금 더 변주를 주어 Perfect, Great, Good, Bad 4개의 세밀한 판정을 내릴 수 있게 했다.

- t.write("문자열", False, "center", ("", 15)) 문장

write 함수에서 False, "center"를 정확히 입력했는지 확인해야한다. 또한, ("", 15)에서 ""는 큰따옴표 두 개를 붙여 쓴 것인데, 내용이 없는 빈 문자열을 의미한다.

:‘거북이 대포’에서는 판정결과를 나타내기 위해 허용된 문장이다.

- 키 이름 입력 문제

t.onkeypress 함수를 호출할 때 두 번째 인자인 "키 이름"을 입력할 때 주의해야한다. 이 프로젝트에서는 "Up", "Down", "space"가 사용되었는데, "Up" 과 "Down"이 대문자로 시작하는 것과 달리 "space"는 전부 소문자이다.

학습 과정에서 Space를 대문자로 입력했더니\_tkinter.TclError : bad event type or keysym... 에러가 발생했다.

<들어가기>

말 그대로 "거북이 대포"이므로 idle에(기존에 사용하던 아톰이 아닌 idle을 사용하였다) turtle을 호출한 후, turtle을 t라고 정의했다. 그 후 turtle의 모양을 "거북이"로 설정했다.(t.shape("turtle"))

또한 left, right, forward 명령으로 대포가 날아가는 모습을 화면에 보여 줬다. 이때 거북이 펜 상태를 up으로 지정하여 거북이 대포가 지나가는 자리에는 선이 그려지지 않는다.

가장 중요한 역할을 하는 fire함수는 스페이스 바를 누르면 실행되도록 코드를 작성했다. 먼저 def  
으로 fire함수가 어떤 역할을 하는지 정의를 해주고,

```
def fire():
```

```
    ang = t.heading()
```

```
    while t.ycor()>0:
```

```
        t.forward(15)
```

```
        t.right(5)
```

t.onkeypress 메소드를 사용하여 space바를 눌렀을 때 바로 거북이가 날아갈 수 있게 정해준다.

함수의 처음과 끝은 각각 거북이(화살촉 모양 대포)의 "현재 각도를 저장하는 부분"인 ang =  
t.heading()과 거북이를 "처음 위치와 각도로 되돌리는 부분"인 t.setheading(ang)로 되어 있다. 이렇  
게 한 이유는 대포가 날아가는 동안 거북이 대포의 위치와 각도가 계속 변하기 때문이다. 따라서  
이 값을 처음 상태로 되돌려 다음 발사를 할 수 있도록 준비하는 것이다.

다음 while 반복문이 바로 실제로 대포를 날아가게 하는 부분이다.

t.ycor()는 현재 거북이(대포)의 y 좌표값을 구하는 문장이다. 땅의 y 좌표값은 0이므로 땅보다 위로,  
즉 하늘로 날아가는 동안 거북이의 y 좌표값은 언제나 0보다 크다. 대포가 날아가다가 땅에 떨어지  
는 순간, 즉 거북이의 y 좌표값이 0보다 작아지는 순간이 바로 대포가 땅에 닿은 순간이다. 이렇게  
거북이가 땅에 닿는 순간 while 반복문을 빠져나오게 되고 대포가 멈추게 되며, 이는 거북이가 움  
직이지 않음을 의미한다..

while 반복문을 빠져나오면 t.distance 함수로 현재 거북이의 위치와 목표 지점(target, 0)과의 거리  
를 구한다. if-else문을 사용하여 목표 지점과의 거리가 15보다 작으면 Perfect!를 출력하고, 25보다  
작으면 Great를, 40보다 작으면 Good을, 불발한 것으로 보고 Bad!를 출력한다. 그 이외의 값은  
"Bad!"가 출력되도록 했다.

### 학습 방법 및 과정

처음에는 팀별로 제작해보려다가 각자의 역량을 살리고, 더 퀄리티 있는 게임을 만들기 위해 모두  
거북이 대포게임을 만들어 보기로 했다. (서기: 김예린 학우)

그러다가 김예린 학우가 만드는 게임에서 오류가 발생했고, 신예슬 학우는 거북이가 날아가는 경로  
설정에 오류가 있다는 것을 발견하여 다음과 같이 제대로 알려주었다.

```
while t.ycor() > 0:
```

```
    t.fd(15)
```

```
    t.rt(5)
```

```
# 거북이가 날아가는 경로 설정
```

```
# 앞으로 가면서 5도씩 회전(포물선 형태는 아니고 원에 비슷한 형태로 날아감. 정확히는 정72각
```

**형 형태)**

그 이후에 강미규 학우의 질문이 이어졌다.

" t.sety(random.randint(10, 100) <- 이게 성공, 실패를 표시할 위치를 지정한다는데 t.sety가 뭐하는 함수인지 모르겠어. 그리고 t.setheading(x) 이건 무슨 뜻이야?"

이정란 학우, 김현지 학우가

먼저 이정란 학우가 "t.sety는 y를 set한다는 건데, 여기서 거북이의 y좌표를 설정하는 거야!" 라고 알려주었고, 다음으로 김현지 학우가 " t.setheading(x)는 거북이의 머리를 x로 설정한다, 란 뜻이니까 거북이 각도를 x로 설정한다는 뜻이야." 라고 대답했다.

**<학습소감>**

[신예슬 학우]: 자바와 씨름하다가 오랜만에 파이썬을 하니깐 어색했다. 자바와 문법도 헛갈리고 새삼 파이썬의 간결한 문법에 놀라기도 했다. 그만큼 자바에 익숙해졌나 싶었고, 드디어 파이썬으로 게임을 만든다고 생각하니 활용을 해보는 것 같아 신났다. 몇 번의 오류 끝에 게임 구현에 성공했다. 하지만 아쉬운 점이 몇 개 보였다. 개선해야 할 기능들이 보였다. 기능을 더 개선하고 싶다는 생각이 계속 들어서 직접 문법을 구글링해서 고치기도 했다. 나도 이제 개발자가 된건가 싶었다. 더 깔끔한 ui와 재밌는 요소들을 추가하고, 보기 안좋은 것들은 삭제하고 싶었다. 예를들면 화면에 Bad 글씨가 사라지지않는 문제를 해결하고 싶었다. 그래서 이런 저런 방법을 생각해보다가 아예 Bad 글자를 출력하는 문장을 삭제해보기도하고, 반복문으로 돌려보기도하고, if문을 좀 더 구체화해보기도 했다. 대포게임은 간단하고 짧은 코딩으로 할 수 있는 게임이지만 그렇기에 더 어렵기도 하고, 수정하고싶은 코드들이 많아서 더 재밌기도했다. 코드를 수정해 원하는대로 구현이 되면 정말 뿌듯했다.

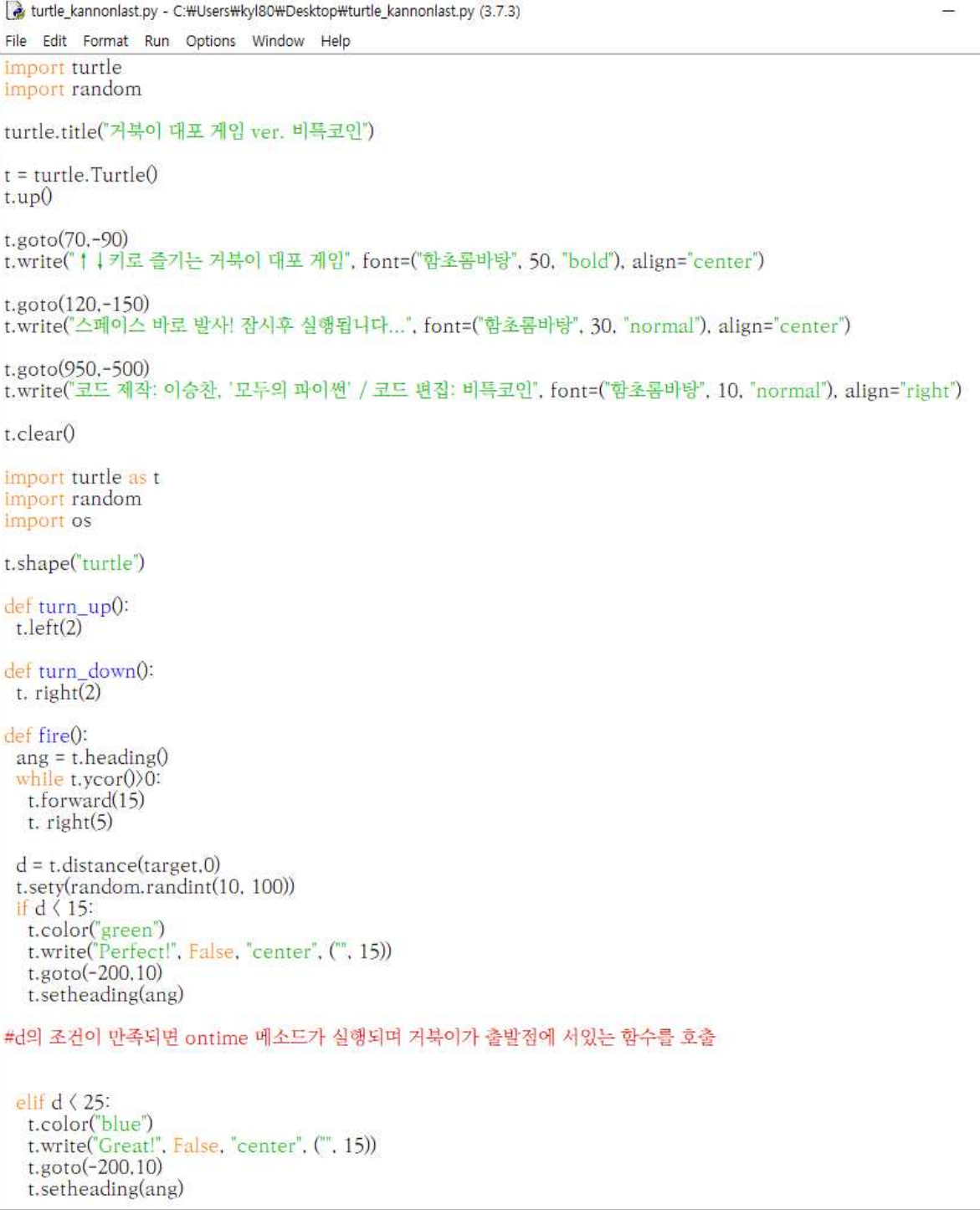
[강미규 학우]: 파이썬은 이미 4주차에 끝내서 한동안 프로그램을 실행하지 않았기 때문에 파이썬으로 게임을 만든다는 게 굉장히 신났다. 사실 총 3번의 에러가 났는데, 처음엔 오류 때문에 게임이 실행조차 되지 않았다. 다행히 컴파일러가 에러를 잡아내어 두 번은 쉽게 수정했고, 마지막엔 무엇이 문제인지 찾아내질 못해서 모든 팀원들에게 코드를 공유했다. 들여쓰기에 문제가 있던 것을 발견하여 수정하고 나니 거북이가 제대로 날아갔다.


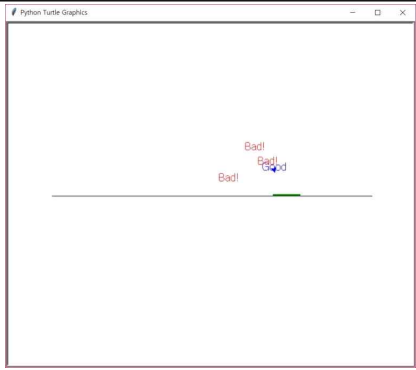

[이정란 학우]: 이론과 간단한 코딩 정도로만 배운 파이썬이었기에 게임을 직접 만드는 것이 처음에는 너무 어렵게 다가왔다. 하지만 대략 한 달 간 배운 파이썬이었고 거북이 대포게임은 매우 간단한 게임이라서 입문용 게임으로 아주 좋았다.

[김예린 학우]: 자바를 배우는 사이에 파이썬을 다 잊어버려서 처음에는 조금 헤맸다. 차라리 자바를 먼저 배우고 파이썬을 나중에 배울걸 싶었다. 하지만 천천히 코드를 작성하고 발생한 오류(거북이가 땅에서 떨어지자 원을 그리며 빙빙 돌, "bad" 판정문구에 계속해서 튕겨져 나감 등)를 해결하

는 과정에서 잊었던 함수들과 모듈 등의 개념, 그리고 몰랐던 함수들을 알게 되어 정말 좋은 기회였던 것 같다. pygame이라는 것도 있다던데 다음엔 그걸 이용해서 아예 제대로 된 게임을 만들어 보고 싶다.

[김현지 학우]: 나는 아무래도 파이썬을 잡고 있던 시간이 길다 보니 파이썬이 되게 반가웠다. 정말 오랜만에 Idle을 켜보니 어떤 것보다도 깔끔한 화면이 반가우면서 뭐라도 적을 수 있을 것 같은 생각이 들었다. 일단 거북이 대포 코드를 다 적고 실행해봤는데 짧은 코드의 어쩔 수 없는 허접함이 재밌었다. 조금 더 깔끔하기 위해 다같이 코드를 수정했다. 실행하고 싶은 것이 있고, 코드를 조금만 수정하면 실행될 것 같은데 어떻게 할지 모르겠는 마음에 더 공부하고 싶다는 생각이 들었다. 검색을 해도 한계가 있었다. 나는 거북이 대포를 두 개 만드는 것을 어떻게 해내서 기뻐다. 작은 프로그램이지만 코드를 추가하고 수정해서 내가 원하는 대로 만들 수 있는 것이 뿌듯했다.

 <pre> import turtle import random  turtle.title("거북이 대포 게임 ver. 비특코인")  t = turtle.Turtle() t.up()  t.goto(70,-90) t.write("↑ ↓ 키로 즐기는 거북이 대포 게임", font=("함초롬바탕", 50, "bold"), align="center")  t.goto(120,-150) t.write("스페이스 바로 발사! 잠시후 실행됩니다...", font=("함초롬바탕", 30, "normal"), align="center")  t.goto(950,-500) t.write("코드 제작: 이승찬, '모두의 파이썬' / 코드 편집: 비특코인", font=("함초롬바탕", 10, "normal"), align="right")  t.clear()  import turtle as t import random import os  t.shape("turtle")  def turn_up():     t.left(2)  def turn_down():     t.right(2)  def fire():     ang = t.heading()     while t.ycor() &gt; 0:         t.forward(15)         t.right(5)      d = t.distance(target, 0)     t.sety(random.randint(10, 100))     if d &lt; 15:         t.color("green")         t.write("Perfect!", False, "center", ("", 15))         t.goto(-200, 10)         t.setheading(ang)  #d의 조건이 만족되면 ontime 메소드가 실행되며 거북이가 출발점에 서있는 함수를 호출  elif d &lt; 25:     t.color("blue")     t.write("Great!", False, "center", ("", 15))     t.goto(-200, 10)     t.setheading(ang) </pre>			
활동 사진 첨부			
학 습 활 동		학 습 결 과	-거북이 대포게임 실행

<p>①</p>		<p>물 ①</p>	
<p>학 습 활 동 ②</p>		<p>학 습 결 과 물 ②</p>	